

REMARKS

Claims 1-12 and 14-19 are currently pending in the application. In an Office Action dated October 03, 2002 ("Office Action"), the Examiner rejected claims 1-6, 8-12, and 16-17 under 35 U.S.C. § 103(a) as being unpatentable over Ofek, U.S. Patent No. 6,101,497 ("Ofek") in view of Breitbart et al., U.S. Patent No. 5,999,931 ("Breitbart"), rejected claims 7 and 18-19 under 35 U.S.C. § 103(a) as being unpatentable over Ofek in view of Lin et al., U.S. Patent No. 5,713,017 ("Lin"), and rejected claims 13-15 under 35 U.S.C. § 103(a) as being unpatentable over Ofek. Applicant's representative respectfully traverses the 35 U.S.C. § 103(a) rejections, below.

Obviousness-Type Rejections

If looked at with the advantage of hindsight, almost any patentable invention may be produced by extracting, modifying, and combining previously existing elements from various devices and processes. The courts have long recognized this, and have repeatedly warned against obviousness-type rejections based on hindsight recognition. For example, the MPEP § 2143.01 states that "[t]he mere fact the references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination." Furthermore, in the same section, the MPEP indicates that a proposed modification cannot render the prior art unsatisfactory for its intended purpose or change its principle of operation. Also, according to MPEP § 2143.03, all claim limitations must be considered in judging the patentability of a claim.

To prevent unreasonable obviousness rejections, the statutes and federal courts have imposed certain conditions and constraints on citing and combining references for obviousness rejections. These constraints are concisely summarized in the first paragraph of MPEP § 2143:

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success.

Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations.

Moreover, according to MPEP § 2144.03:

When a rejection is based on facts within the personal knowledge of the examiner, the data should be stated as specifically as possible, and the facts must be supported, when called for by the applicant, by an affidavit by the examiner.

Rejection of Claim 1 Under 35 U.S.C. § 103(a)

In rejecting claim 1, the Examiner states that Okef teaches "splitting the object and the mirror for the object so that the mirror becomes a backup copy of the object," citing column 9, lines 10-18 of Okef. Okef does not teach splitting a mirror in order to create a backup copy of a data volume. A backup copy is a snapshot of data, in time, that is not subsequently altered, but that is instead archived for possible use in the case of corruption of the live database. While the current application is related to disk-based mirrored LUNs within disk arrays, the current application contains a rather detailed discussion of the concept of backups, beginning on line 12 of page 4. A problem addressed by the present invention is introduced beginning on line 19 of page 6: "Unfortunately, disk-based backup copies may be corrupted by subsequent I/O write operations directed to the LUN on which the backup copy is stored." The present invention is directed to verifying split mirrored logical devices for use as backups for restoring a logical device. Verification involves determining whether or not WRITE operations have been directed to the backup following the mirror split operation. If such WRITE operations have occurred, the backup copy is unusable.

In a section entitled "Independent Operating Mode," beginning on line 23 of Okef, Okef describes operation of the local and remote database systems following a database-mirror split. On lines 24-27 of column 11, Okef states that "[i]n accordance with this invention, it is possible for the host system 40 in FIG. 1 to operate independently with the capability of writing information to the storage device sets 42 and 43." On lines 39-42 of column 11, Okef states that "any writing operation or updating operation that now occurs in the local system 10 still alters data in the storage device sets 15 and 16 in step 62 in FIG. 3." In other words, as stated throughout Okef, included in the cited lines from column 9, Okef's

technique is directed to allowing a mirrored database to be split, and both sides of the mirrored database to be then written to independently. There is no teaching, mention, or suggestion in Okef of splitting the mirrored database for use as a backup copy of the database. Okef is directed to an almost opposite technique – that of providing for independent writing to both sides of a split database. As noted above, even a single WRITE directed to a split mirrored logical device used as backup can irretrievably corrupt the backup and prevent it from being used for restoring a subsequently failed logical device. The present invention is directed to detecting such WRITE operations in order to prevent use of a split mirrored logical device as a backup, while Okef seeks to allow WRITE operations to be performed on both sides of a split, mirrored database. Thus, the Examiner has not cited a reference for the step of "splitting the object and the mirror for the object so that the mirror becomes a backup copy of the object." For this reason alone, claim 1 can not possibly be made obvious by Okef and Breitbart, because neither cited reference teaches or suggests a claim element.

The Examiner acknowledges that Okef does not teach employing a timestamp mechanism to sense corruption of the backup copy, and cites Breitbart as teaching such use of a timestamp. First, the Examiner has made no showing that either Breitbart or Ofek suggests combining a timestamp mechanism employed in a transaction-processing database-management system, disclosed in Breitbart, with a mirrored database system, disclosed in Okef, then modifying the resulting combination to meet the claim-language constraints of claim 1, and finally applying the modified combination to the entirely different logical-device-based mirroring system, as clearly claimed in claim 1. As discussed in the previous subsection, that showing is required. If an explicit showing cannot be made, and the Examiner wishes to rely on personal knowledge of the obviousness of combining the two references, modifying them substantially, and then applying them to an entirely different type of device and system, then Applicants' representative respectfully requests that the Examiner supply an appropriate affidavit, as specified by MPEP § 2144.03.

Second, please consider carefully the cited section of Breitbart. Breitbart associates a timestamp with a transaction – "the transaction is assigned a timestamp" (Breitbart, column 3, line 42). Note that a transaction is a set of one or more I/O operations – generally READ and WRITE operations. Then, during execution of a transaction T_i , the timestamp of the transaction is compared with that of any other transaction that executed a

WRITE operation to any data item to which a WRITE operation included in the transaction T_i is directed: "When T_i submits a write operation on the secondary copy of a data item, the timestamp of is compared to that of the transaction which executed the last write operation [to] the same data item" (Breitbart, column 3, lines 47-51). Finally, "[i]f the T_i timestamp is less than that of the last write on the data item, the T_i write operation is not performed" (Breitbart, column 3, lines 51-53). Thus, in Breitbart, rather ephemeral transactions are assigned timestamps which are compared in a "less than" operation to timestamps assigned to other transactions now associated with individual data items in a database. Please compare this technique with Applicants' claim 1, provided below:

1. A method for backing up a computer-readable object stored on a first logical device unit, the method comprising:
 - when the object is not currently mirrored to a mass storage device, creating a mirror for the object on a second logical device unit on a mass storage device;
 - when the object and the mirror for the object are split, resyncing the object with the mirror for the object;
 - splitting the object and the mirror for the object so that the mirror becomes a backup copy of the object and so that I/O requests directed to the object are not automatically directed to the mirror;
 - retrieving a current timestamp from the second logical device and saving it as a saved timestamp;
 - updating the timestamp upon executing any I/O operation directed to the second logical device that alters data stored on the second logical device;
 - when the object is determined to need to be restored from the mirror,
 - retrieving a current timestamp from the second logical device;
 - comparing the retrieved current timestamp to the saved timestamp;
 - when the current timestamp is equal to the saved timestamp, copying the mirror to the first logical device to replace or again create the object on the first logical device. (emphasis added)

Please note the many differences between the use of timestamps in the cited reference, and the claimed invention. Breitbart assigns timestamps to transactions, essentially I/O operations. By contrast, Applicants' claimed method retrieves a current timestamp associated with a logical device. Breitbart maintains a presumably large set of transactions that include timestamps in association with individual data items. By contrast, in Applicants' claimed method a logical device, containing generally thousands to billions of data items, maintains a current timestamp by Applicants' method. The final series of steps in claim 1 are performed "when an object is determined to need to be restored from the mirror." Neither Breitbart nor

Ofek teaches or mentions restoring an object from a split mirrored logical device used as a backup copy of the logical device. In this final series of steps, Applicants' method again retrieves a current timestamp from a logical device. Breitbart does not teach or mention a current timestamp maintained with logical devices. Finally, Applicants' invention determines whether the current timestamp retrieved from a logical device is *equal* to a previously saved timestamp retrieved from the same logical device. Breitbart compares timestamps assigned to transactions, in a "*less than*" operation, to timestamps associated with data items. Breitbart's technique is directed to "generate a globally serializable transaction schedule" (Column 3, lines 18-19). Applicants' invention is directed, by contrast, to determining whether a split mirrored logical device used as a backup has been corrupted. In general, the term "logical device" appears not to be considered by the Examiner when reading the claim elements unto the cited references. A logical device is not a database management system, nor a networked file system.

Applicants' representative readily acknowledges that the art is filled with references to database mirroring and to timestamps. However, Applicants' representative has no knowledge of a current timestamp being associated with a logical device in a mass storage device, such as a disk-array, for use in evaluating split mirrors for suitability for restoration of corrupted or inaccessible logical devices. The cited references appear to have no relevance to such a combination, failing to teach or suggest a combination of mirroring techniques and timestamps, failing to teach or mention use of split mirrors for backups, and failing to teach or suggest the majority of claimed elements in claim 1. Claim 1 is not obvious in view of Ofek or Breitbart alone, or in combination.

Rejection of Claim 1 Under 35 U.S.C. § 103(a)

As with the rejection of claim 1, the Examiner cites column 9, lines 10-18 of Ofek as teaching "splitting the object and the mirror for the object so that the mirror becomes a backup copy of the object," but, as discussed above, Ofek does not teach or suggest anything related to using a split mirrored logical device as a backup.

The cited section of Lin also does not teach Applicants' claimed invention. Consider claim 7, below:

7. A method for backing up a computer-readable object stored on a first logical device unit, the method comprising:
 - when the object is not currently mirrored to a mass storage device, creating a mirror for the object on a second logical device unit on a mass storage device;
 - when the object and the mirror for the object are split, resyncing the object with the mirror for the object;
 - splitting the object and the mirror for the object so that the mirror becomes a backup copy of the object and so that I/O requests directed to the object are not automatically directed to the mirror;
 - retrieving a current count from the second logical device and saving it as a saved count;
 - incrementing the count executing any I/O operation directed to the second logical device that alters data stored on the second logical device;
 - when the object is determined to need to be restored from the mirror,
 - retrieving a current count from the second logical device;
 - comparing the retrieved current count to the saved count;
 - when the current count is equal to the saved count, copying the mirror to the first logical device to replace or again create the object on the first logical device.

Again, neither Lin nor Ofek teaches or suggests anything with regard to splitting a mirrored logical device for use as a backup. Lin's sequence server is responsible for "assigning a unique sequence number for each command received by the server group" (Column 7, lines 22-24). By contrast, Applicants' claimed system retrieves a current count maintained by a logical device from a logical device. In the cited section of Lin, Lin describes comparing a "sequence number, Mi, stored in the message" with an I_Counter (Lin, column 7, lines 61-63). If the sequence number "Mi is larger than the I_Counter by 1, then the server will update its local file system and increment the 'I_Counter' to indicate that the update request has been serviced. If Mi is equal to or less than the I_Counter, the update request is a duplicate and discarded" (Lin, column 7, lines 63-67). Applicants' system, by contrast, involves comparing, for *equality*, a current counter associated with a logical device to a stored counter previously obtained from the logical device. Finally, Lin is directed to serializing updates in a distributed networked file system: "With the sequence numbers and the first and second counters, updates will be performed in the same order on different file servers" (Abstract, last sentence). Applicants' invention is directed, by contrast, to determining whether a split mirrored logical device used as a backup has been corrupted.

There is no similarity with Lin's disclosed technique and Applicants' claimed invention, other than the use of counters, but, as with timestamps, Applicants' representative readily acknowledges the prior existence of counters. Neither Lin nor Ofek even remotely suggests a combination of an update technique for serializing updates in a distributed networked file system with a database mirroring system, modifying the combination to produce a system containing the elements of claim 7, and then applying the combination to mirrored logical devices for ascertaining the reliability of a split mirror for use as a backup. Neither Lin, Ofek, nor a combination of Lin and Ofek teach or suggest the majority of claimed elements of claim 7. Therefore, claim 1 cannot be made obvious by Lin or Ofek, alone or in combination.

Rejection of Dependent Claims 2-6 and 8-12 Under 35 U.S.C. § 103(a)

Because the independent claims 1 and 7 are not made obvious by Lin, Breitbart, Ofek, or any combination of Lin, Breitbart, and Ofek, dependent claims 2-6 that depend from claim 1 and dependent claims 8-12 that depend from claim 7 are not made obvious by Lin, Breitbart, Ofek, or any combination of Lin, Breitbart, and Ofek. The Examiner makes a number of statements asserting that Ofek and Breitbart teach various elements of the dependent claims related to backups and timestamps associated with logical devices, but, as discussed above, using a split mirrored logical device as a backup is neither taught nor suggested in any of the cited references, and Breitbart's timestamps are associated with transactions and are employed for serializing transactions, an application of timestamps to a system and problem space completely different from and unrelated to Applicants' claimed invention.

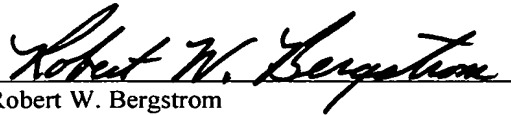
Rejection of Claims 13-19 Under 35 U.S.C. § 103(a)

Applicants' representative has cancelled claim 14, and amended claim 13 to include the elements of originally filed claim 14. Claims 13 and 15-19 are directed to: "A mass storage device that provides logical device units to accessing computers, the mass storage device comprising: a medium for storing data; data writing and reading devices for writing data to the medium and reading data from the medium; memory and logic processing components; and a controller that executes within a logic processing component" (newly

amended claim 13). Ofek is directed to a distributed database in which mirrored databases can be split and independently operated. In fact, Ofek explicitly states that each of the local and remote systems over which the database is distributed "may comprise a Symmetrix integrated cached disk array ... Consequently, the following discussion makes only general references to the operation of such systems. For the purposes of this invention, it is sufficient to understand that the remote system 11 normally acts as a mirror to of the local system 10 on a volume-by-volume basis and that the volumes can be physical volumes, although logical volumes are preferred" (Column 8, lines 46-56). In other words, Ofek's disclosure is completely unrelated to the workings and internal components of mass storage devices, such as disk arrays. Claim 13 explicitly claims a technique for verifying split mirrored logical units for use as backups within the controller of a mass storage device, such as a disk array. By contrast, Ofek is directed to a mirrored database that can be split and each part of the mirror then used independently from the other. Ofek's disclosure is unrelated to the invention claimed in claims 14-19, as explicitly stated in Ofek. There is no basis for an obviousness type rejection of claim 14 based on Ofek. Because claim 13 is not obvious in view of Ofek, dependent claims 15-19 are also not obvious in view of Ofek or in view of Ofek and any combination of Lin and Breitbart.

All of the claims remaining in the application are now clearly allowable. Favorable consideration and a Notice of Allowance are earnestly solicited.

Respectfully submitted,
Robert Cochran and Harald Burose
OLYMPIC PATENT WORKS PLLC


Robert W. Bergstrom
Registration No. 39,906

Enclosures:

Postcard

Transmittal in duplicate

Version With Markings to Show Changes Made

In the Claims:

Claims 14 is cancelled.

Claims 13 is amended as follows:

13. (amended) A mass storage device that provides logical device units to accessing computers, the mass storage device comprising:

a medium for storing data;

data writing and reading devices for writing data to the medium and reading data from the medium;

memory and logic processing components; and

a controller that executes within a logic processing component and controls reading and writing of data to and from the memory and to and from the medium, the controller providing, in [a] addition to execution of I/O operations, including execution of read and write operations[,] to and from logical device units comprising portions of the medium for storing data, mirroring of an object stored on a first logical device unit to a mirror object stored on a second logical device unit and a current state metric for each logical device unit that can be requested by an accessing computer, the controller updating the current state metric for a logical device unit whenever the controller executes an I/O operation that changes the data, stored on the medium for storing data, included in the logical device unit's data.